

IN THE CLAIMS:

1. A method for enabling JAVA applications to connect object linking-embedding database (OLE DB) via utilization of JAVA Database Connectivity (JDBC) interface, comprising the steps of:
- 5 (a) initiating a JAVA application to contact a JAVA Database Connectivity (JDBC) Driver;
- (b) bridging, by said JDBC Driver, to an OLE DB data provider, to provide an interface between said JDBC Driver and said OLE DB database.
- 10

10071515-020702

2. The method of claim 1 where step (b) includes the steps of:

(b1) providing a series of JAVA classes that implement the JDBC interface;

5 (b2) implementing a Datalink Library in C++ for the JAVA classes to act as an OLE DB client for said OLE DB data provider.

3. The method of claim 2 wherein step (b2) includes the step of:

(b2a) connecting said JDBC application to said OLE DB application to establish an OLE DB data provider.

5

4. A method for enabling a client using JAVA applications to access an OLE DB database by the utilization of a JDBC API interface comprising the steps of:

5 (a) establishing, for each interface in the JDBC API, a corresponding JAVA class and a corresponding C++ class;

(b) maintaining in each JAVA object a reference to its corresponding C++ object;

10 (c) maintaining in each C++ object a reference to the OLE DB object which most closely matches the functionality of the JDBC API interface;

(d) passing a database query to said OLE DB database.

5. The method of claim 4 wherein step (d) includes the steps of:

(b1) calling, by a selected JStatement object, for an Execute function on its corresponding C++ Statement object;

(b2) calling, by said C++ Statement object, on the functions of two different interfaces of the corresponding OLE DB Command Object which (i) sets up the text for the query, and (ii) provides an Execute command to execute the query.

6. The method of Claim 5 which includes the step of:

(c) connecting to the OLE DB database to garner the accessed response to the query;

(d) returning the resultant response to said client.

7. The method of claim 4 wherein step (b2) includes the steps of:

(b2a) creating, by said CStatement object, a CResultSet object;

(b2b) making said CResultSet object enable a reference to a Rowset object;

(b2c) passing, by said CStatement, a connecting reference to the newly-created CResultSet object for passing said CResultSet object back to said JStatement object;

(b2d) creating, by said JStatement object, of a new JResultSet object;

(b2e) making said new JResultSet object reference the corresponding CResultSet object;

(b2f) returning, by said JStatement object, of the JResultSet object to said client.

8. The method of claim 7 wherein step (b2f) includes the step of:

(b2f1) utilization, by said Client, at the JResultSet object to access the data returned by said database in response to said query.

9. The method of claim 7 wherein step (b2f) includes the step of:

(b2f2) implementing the said ResultSet interface, as defined by the JDBC standard, to utilize said JResultSet object to access the data returned from the database in response to said query.

5

20071846-020702

10. A system for enabling a client utilizing a JAVA API application to access a database which is designed for communication with OLE DB-type interfaces comprising:

5 (a) JDBC Driver means that utilize JDBC API JAVA standards and perform a bridge interface operation between said JAVA API application and an OLE DB data provider which implements the OLE DB API specification;

10 (b) means to pass a database query initiated by said JAVA (JDBC API) client to a database organized for OLE DB API clients and receive a response to said database query suitable for said JAVA JDBC API application to receive properly.

11. The system of claim 10 wherein said JDBC Driver means includes:

(b1) a series of JDBC API interfaces wherein each said JDBC interface contains a JAVA class and a corresponding C++ class;

(b2) means to maintain a reference between each JAVA object and its corresponding C++ object;

(b3) means to maintain a reference between each C++ object and the particular OLE DB object which most closely matches the functionality of the JDBC API interface.

12. The system of claim 11 wherein each C++ object maintains a connective reference to multiple interfaces defined by the said OLE DB object.



13. The system of claim 10 wherein said means (b) to pass a database initiated by said JAVA (JDBC API) client to a database organized for OLE-DB API clients, includes:

- 5 (b1) means to create a JDriver object;
- (b2) means to create a CDriver object;
- (b3) means to make said JDriver object reference itself to said CDriver object;
- (b4) means to create a Data Source object;
- 10 (b5) means to make said CDriver object reference said Data Source object;
- (b6) means to develop a OLE DB Session object;
- (b7) means to make said CSession object reference said OLE DB Session object;
- 15 (b8) means to create a JSession object;
- (b9) means to make said JSession object reference said CSession object;
- (b10) means to return said JSession object to said Java client.
- 20

14. The system of claim 13 which further includes:

- (a) means to create a Statement object, said means including;
- (c1) means to query whether the OLE DB Data Provider supports an OLE DB Command object;
- 5

(c2) if not, then activation means to create a CStatement object which is referenced to said OLE DB Session object.

10071815 020702

15. The system of claim 13 wherein if said OLE DB Data Provider does support said OLE DB Command object, then said activation means includes:

(c2a) means to create an OLE DB Command object;

(c2b) means to create a CStatement object which is referenced to said OLE DB Command object;

(c2c) means to create a JStatement object which is referenced to said CStatement object;

(c2d) means to return said JStatement object to said client.

16. The system of claim 15 wherein said means to create an OLE DB Command object includes:

(d1) means to create a CPreparedStatement object or a CCallableStatement object which is referenced to said Command object;

(d2) means to create a JPreparedStatement object or JCallableStatement object which is respectively referenced to said CPreparedStatement object or said CCallableStatement object;

(d3) means to return, to said client, said JPreparedStatement object or said JCallableStatement object.

17. The system of claim 14 wherein said means (c1) to query whether said OLE DB Data Provider supports an OLE DB Command object indicates a negative (NO) response, includes:

5 (d3) means to catch this negative response as an exception and display an error code.

18. The system of claim 15 wherein said means (c2b) to create a CStatement object referenced to said OLE DB Command object includes:

5 (e1) means to indicate that said CStatement object does reference said OLE DB Command object;

(e2) means to execute said OLE DB Command of said Command object;

10 (e3) means to question if said OLE DB Command object involves a query;

(e4) means, if said Command object involves a query, to create a Rowset object;

15 (e5) means to create a CResultSet object which is referenced to said Rowset object;

(e6) means to create a JResultSet object which is referenced to said CResultSet object;

20 (e7) means to return to said client, said JResultSet object.

19. The system of claim 17 wherein said means (c2b) to create a CStatement object referenced to said OLE DB Command object includes:

5

(ey1) means to indicate that said CStatement object does not reference said OLE DB Command object;

(ey2) means to indicate that the command to said OLE DB Command object is a simple query;

10

(ey3) means to call the Open Rowset function of the IOpenRowset interface which is supported by the Session object;

15

(ey4) means to create a Rowset object;

(ey5) means to create a CResultSet object referenced to said Rowset object;

20

(ey6) means to create a JResultSet object referenced to said CResultSet object;

(ey7) means to return to said client, said JResultSet object.

10074815.000702

20. The system of claim 14 wherein said means (c2) to create a CStatement object referenced to said OLE DB Command object includes:

- (i) means to indicate that said CStatement object does not reference said OLE DB Command object;
- (ii) means to indicate that the command to said OLE DB Command object is NOT a simple query;
- (iii) means to flag an exception condition and display an error message.

21. The system of claim 15 wherein said means (c3) to question said OLE DB Command object indicates that it is NOT a query, then said system includes:

- (e7) means to return to said client, a count of the number of records which have been updated.

22. The system of claim 15 wherein said means (c2a) to create an OLE DB Command object includes:

- (f1) means to execute an SQL command;
- (f2) means to indicate said command is a query;
- (f3) means to create an OLE DB Rowset object to handle said query;
- (f4) means to create a C++ ResultSet object which references said Rowset object;
- (f5) means to create a JAVA JResultSet object which references said CResultSet object created in clause (f4);

15 (f6) means to return said `JResultSet` object to the client application.

23. The system of claim 22 wherein means (f4) to create said `CResultSet` object includes:

(g1) means to create a `CResultSetMetaData` object;

5 (g2) means to acquire column information from said `Rowset` object;

(g3) means to create a `JResultSetMetaData` object based on said column information;

10 (g4) means to return to said client, a reference to said `JResultSetMetaData` object.

24. The system of claim 23 wherein said means (g1) to create a `CResultSetMetaData` object includes:

5 (h1) means to create a `CDatabaseMetaData` object which is referenced to said `OLE DB Session` object;

(h2) means to create a `JDatabaseMetaData` object which is referenced to said `CDatabaseMetaData` object;

10 (h3) means to return to said client, said reference to said `JDatabaseSetMetaData` object.

25. The system of claim 22 wherein said means (f3) to create an OLE DB Rowset object includes:

(d1) means to create a CResultSet object which is referenced to said OLE DB Rowset object;

5 (d2) means to create a JResultSetMetaData object based on said column information((g2) claim 13);

(d3) means to return to said client, a reference to said JResultSetMetaData object.

10071845-020702